Chapter 5

Adaptive multivariate degradation model for remaining useful life prediction

Adriana Villalón Falcón[1], Alberto Prieto Moreno[1], Marcos Quiñones-Grueiro[2], and Orestes Llanes-Santiago[1]

Abstract Accuracy in predicting the remaining useful life (RUL) of industrial systems is crucial for maintenance tasks. Deep learning-based methods are among the most widely used for RUL prediction because of their ability to learn complex non-linear relationships between system measurements and RUL. After training, however, these models are incapable of adapting to systems with dynamic behavior that changes over time. On the other hand, prognostic methods based on degradation models, while capable of adapting to these changes, are usually designed to model a single variable. In this chapter, an adaptive method for predicting RUL based on modeling the behavior of multiple variables during degradation is proposed. The information from each model is combined to predict the RUL of the system. A new performance metric is proposed to evaluate the prediction models by periodically calculating the prediction error and establishing a direct relationship between RUL prediction and maintenance planning. The proposed method is applied to the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) data set, and demonstrates satisfactory results.

Nomenclature

$\alpha_{RUL_{i,t}}$	weight of RUL (remaining useful life) prediction error of system i at time t
CDF^{-1}	inverse cumulative distribution function
EOL_i	true EOL (end of life) of system i
$EOL^*_{j,i,t}$	predicted EOL with feature j of system i at time t
ET_j	EOL threshold of feature j
f_j	degradation model of feature j
N	number of systems
$\mathbf{p}$	vector of parameters of f
$\mathbf{p}_L, \mathbf{p}_U$	lower and upper bounds of parameters $\mathbf{p}$
R_s	constrained region of the parameter space
$RUL_{i,t}$	true RUL of system i at time t

[1]Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE, La Habana, Cuba,
e-mail: avillalon@automatica.cujae.edu.cu, albprieto@automatica.cujae.edu.cu and orestes@tesla.cujae.edu.cu
[2]Institute for Software Integrated Systems, Vanderbilt University, Nashville, Tennessee, USA, e-mail: marcos.quinones.grueiro@vanderbilt.edu

$RUL^*_{i,t}$	predicted RUL of system i at time t
$RUL^*_{j,i,t}$	predicted RUL with feature j of system i at time t
RUL_w	value of RUL from which the precision of the RUL prediction is considered critical
$RWMSE$	global prediction error of a set of systems
$RWMSE_i$	global prediction error of system i
t_d	detection time of the degradation process
T	number of observations
V	number of features
w_j	weight of feature j
y_j	feature j

5.1 Introduction

Deciding the amount of time that industrial facilities can continue operating and maintaining quantitative and qualitative production standards is generally based on corrective or preventive maintenance strategies. Both maintenance approaches considerably increase production costs and product delivery times. Predictive maintenance, on the other hand, aims to minimize the downtime of systems and maximize their useful life by predicting when a system cannot operate satisfactorily anymore (Niu, 2017; Kim et al., 2016; Lei et al., 2018).

The planning of the predictive maintenance tasks and their correct execution to a large extent depends on the prediction of the RUL. Because of the large amount of data provided by data acquisition systems, many methods capable of providing accurate predictions of RUL from measurements of process variables have been developed.

Among the most widely used data-driven methods are deep neural networks (DNNs), because of their ability to model complex nonlinear relationships (Ellefsen et al., 2018; Guo et al., 2017; Li et al., 2018; Yang et al., 2019; Zheng et al., 2017; Wu et al., 2018). Some network architectures have been designed to consider temporal features that reflect the degradation of systems (Ellefsen et al., 2018; Zheng et al., 2017; Wu et al., 2018). Configuring a DNN takes time, however, and is computationally expensive. After training, a fixed model is obtained that cannot adapt to the system changing conditions over time. If system degradation exhibits new behavior, the DNN must be trained again.

Methods based on degradation models predict the RUL by adapting to the systems' conditions during the degradation process (Li et al., 2015; Feng et al., 2016; Lei et al., 2016; Ali et al., 2018; Saidi et al., 2017; Yan et al., 2020). These methods only model one variable. The prediction of the RUL, however, is not determined by one variable but by the simultaneous evolution of multiple variables. Therefore, multivariable analysis is required. This chapter proposes an adaptive method for predicting the RUL based on modeling the behavior of features that provide significant information during the degradation process. The information from each model is considered for predicting the RUL of the system.

An important element in the configuration of prognostic systems is the selection of the adequate models to predict the RUL because of the constant development of new prediction methods designed for different applications. Although sometimes the selection of a method is based on a particular application, often it is necessary to compare several models according to their performance in the prediction

task. Therefore, performance metrics are used to assess the RUL prediction error. The most commonly used performance metrics are related to the accuracy and precision of prognostic methods (Saxena et al., 2008; Lei et al., 2018; Zeng et al., 2017). This chapter will focus on accuracy metrics, given their importance in prognostic performance. A wide range of metrics have been developed to measure the accuracy of prognostic methods. Accuracy metrics quantify the similarity between the model prediction and true measured values (Zeng et al., 2017). Generally, metrics compute the similarity as the difference between the RUL predicted values (RUL^*) and true values (RUL), Δ, also called prediction error (5.1):

$$\Delta = RUL^* - RUL. \tag{5.1}$$

Accuracy metrics are created by modifying (5.1) to add desired features for metrics based on the capability of prognostic methods for supporting maintenance decisions. The most relevant features that metrics include, ordered by their importance, are detailed below:

1. *Overall performance*. Accuracy must be measured over the lifetime of a system, capturing the prediction error behavior. Some metrics have been used to measure the prediction error in an instant of the system degradation. RUL prediction is a continuous process; thus, the evaluation of methods for this purpose requires measuring how the prediction error changes over time. It is not enough to consider the error at specific time instants as a unique evaluation measure.
2. *Metric value in time units*. Accuracy values must be defined in time units according to the RUL measurement (i.e., hour, day, cycle). The time unit of the prediction is vital for measuring the accuracy of prognostic methods, which allows establishing a connection to equipment operation and maintenance planning. Some metrics provide normalized values, generally in the range $[0,1]$, which mask the prediction time unit. Measures such as mean and median are commonly used to quantize the prediction error over a time period.
3. *Time-based penalization*. Decision-making is critical towards the end of life of a piece of equipment. Therefore, a penalization factor must be added to attribute greater importance to prediction errors made near the end of life. Usually, a weight function penalizes the error given the time instant at which the prediction is made. A linear function (5.2) and a Gaussian kernel (5.3) have been used as penalization functions (Saxena et al., 2008):

$$\alpha(t) = \frac{t}{\sum_{t=t_0}^{T} t}, \tag{5.2}$$

   where t_0 and T are the start and end of life, respectively, and

$$\alpha(t) = \exp\left(-\frac{(t-T)^2}{2(\frac{T}{2})^2}\right). \tag{5.3}$$

4. *Late prediction-based penalization*. Late predictions (positive Δ) are penalized more than early predictions (negative Δ) because of the impact on maintenance. Positive prediction errors are made by predicting a larger RUL value than the true RUL, causing the system to reach the end of life before maintenance is scheduled. Conversely, negative prediction errors favor the execution of early main-

tenance. Commonly, a function that receives a prediction error and retrieves a value related to its magnitude and sign is used. An exponential function (5.4) has been used as a penalization function:

$$\zeta(t) = \begin{cases} \exp\left(-\frac{\Delta}{\varphi_1}\right) - 1, \text{ if } \delta < 0 \\ \exp\left(\frac{\Delta}{\varphi_2}\right) - 1, \quad \text{if } \delta \geq 0 \end{cases}. \tag{5.4}$$

A summary of how these features are included in the most relevant accuracy metrics is presented in Table 5.2.

Table 5.2: Features of accuracy metrics

Metric	Overall Performance	Value in Time Units	Time Penalization	Late Prediction Penalization	References
Mean Absolute Error (MSE)	Yes	Yes	No	No	(Zhang et al., 2019; Zhu et al., 2018)
Exponential Transformed Accuracy (ETA)	No	Normalized	No	Yes	(Lei et al., 2018)
Relative Accuracy (RA)	No	Normalized	No	No	(Saxena et al., 2008)
Cumulative Relative Accuracy (CRA)	Yes	Normalized	Yes	No	(Saxena et al., 2008)
Mean Squared Error (MSE)	Yes	Yes	No	No	(Zeng et al., 2017; Di Maio and Zio, 2016)
Root Mean Squared Error (RMSE)	Yes	Yes	No	No	(Zeng et al., 2017; Di Maio and Zio, 2016; Lei et al., 2018)
Mean Absolute Percentage Error (MAPE)	Yes	Normalized	No	No	(Zeng et al., 2017; Di Maio and Zio, 2016; Zhang et al., 2019)
Sample Mean Error (SME)	Yes	Yes	No	No	(Zeng et al., 2017; Di Maio and Zio, 2016)
Sample Median Error (SMeE)	Yes	Yes	No	No	(Zeng et al., 2017; Di Maio and Zio, 2016)
Timeliness Weighted Error Bias (TWEB)	Yes	Normalized	Yes	Yes	(Zeng et al., 2017; Di Maio and Zio, 2016)

A key element for performance measurement and selection of RUL prediction methods is their accuracy as systems approach to the end of their useful life. Thus, time-based penalization is an important feature in accuracy metrics. Although penalization functions presented in (5.2) and (5.3) increase the penalization as time passes, they do not include information about the criticality of the maintenance task according to the specific application. Therefore, a new accuracy metric for evaluating the performance of RUL prediction methods, which includes the first three features presented in Table 5.2 and captures knowledge for supporting decision-making, is proposed in this chapter.

The first main contribution of this chapter is the proposal of a multivariate degradation model for predicting the RUL, which adapts to system conditions. In the construction of the model, only the variables that contribute significantly to the prognostic task are used. In addition, the reconfiguration of the prognostic model does not require a large amount of data. The second main contribution of this chapter is the proposal of a new performance metric for RUL prediction models that evaluates the prediction error throughout the degradation process in time units and considers task-specific knowledge for decision-making support.

The organization of the chapter is as follows. In Sect. 5.2, the new metric for the comparison of RUL prediction models is presented. Sect. 5.3 details the proposed method for predicting RUL. Sect. 5.4 validates the proposals by using the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) benchmark (Saxena et al., 2008). Finally, the conclusions and several considerations for future research are presented.

5.2 Performance metric

Predictive maintenance tasks should be performed when the longest operating time of a system has been reached and before the end of its useful life to avoid unexpected stops of the production lines. Therefore, RUL prediction methods should be focused on minimizing the prediction error, fundamentally when a system approaches the end of its useful life. Thus, this chapter proposes a metric that periodically considers the error during the degradation process and includes time-based penalization, while keeping error values in time units.

5.2.1 Proposed performance metric

The proposed metric is defined below.

Definition 5.1. *Root Weighted Mean Squared Error*

Given a set of representative systems of the same type of system $\mathbf{E}$, $\{E_i \in \mathbf{E} : i = 1,2,...,N\}$ with cardinality N and each system consisting of a measurement set of size m with the respective time-stamped remaining useful life labels ($E_i = \{Y_i \in \Re^{m \times T}, RUL_i \in \Re^T\}$), the **Root Weighted Mean Squared Error** (*RWMSE*) for $\mathbf{E}$ is obtained as follows:

$$RWMSE = \frac{\sum_{i=1}^{N} RWMSE_i}{N}, \tag{5.5}$$

where

$$RWMSE_i = \sqrt{\frac{\sum_{t=1}^{T} \alpha_{RUL_{i,t}} \left(RUL^*_{i,t} - RUL_{i,t}\right)^2}{\sum_{t=1}^{T} \alpha_{RUL_{i,t}}}} \tag{5.6}$$

is a weighted average of the prediction errors during the degradation of system i, $RUL^*_{i,t}$ and $RUL_{i,t}$ are the predicted and true RULs at instant t during the degradation of system i, respectively; T is the total time over which the measurements were obtained; and $\alpha_{RUL_{i,t}}$ is the weight assigned to the RUL prediction error of system i at instant t.

The basis for this metric is to find a way to determine $\alpha_{RUL_{i,t}}$. This factor should penalize more the prediction error as the system approaches the end of its useful life and less when it is further away from

ceasing to work properly. Several types of functions could be used to determine $\alpha_{RUL_{i,t}}$. In this chapter, the exponential function given in (5.7) is selected, where a, b and c are the parameters of the function. This function places greater weight on prediction errors near the end of the systems life:

$$\alpha_{RUL_{i,t}} = a\exp(bx) + c. \tag{5.7}$$

The next step is to determine the parameters a, b and c of this function. In order to establish a relationship between $\alpha_{RUL_{i,t}}$ and $RUL_{i,t}$, x is defined according to the following equation:

$$x = RUL_{i,t} - RUL_w, \tag{5.8}$$

where RUL_w (w for warning) is the value of RUL from which the precision of the RUL prediction is considered critical, so the weight assigned to the error is greater from that moment on (see Fig. 5.1). In practice, this value can be defined by experts, or it can be assigned based on the knowledge acquired from data. The behavior of $\alpha_{RUL_{i,t}}$ corresponds to

$$\begin{cases} \alpha_{RUL_{i,t}} < a + c, \text{ if } RUL_{i,t} > RUL_w \\ \alpha_{RUL_{i,t}} = a + c, \text{ if } RUL_{i,t} = RUL_w \\ \alpha_{RUL_{i,t}} > a + c, \text{ if } RUL_{i,t} < RUL_w. \end{cases} \tag{5.9}$$

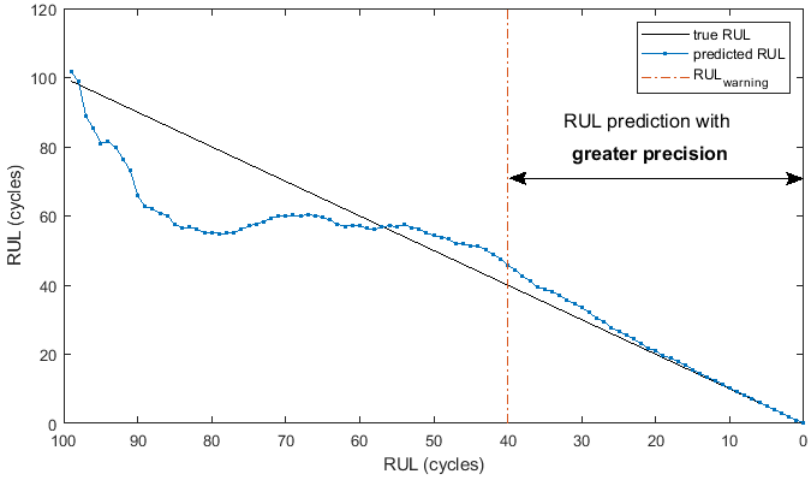


Fig. 5.1: Region of greater precision in the RUL prediction

Conditions on the parameters are $a \geq 0$ and $c = 0$ such that the effects of errors do not cancel each other and that each error contributes to the average, regardless of the time remaining from the prediction instant to the system end of life. Moreover, $a = 1$ to differentiate the weights assigned to both sides of RUL_w. By substituting the values assigned to the parameters in the inequalities (5.9), it is verified that the stated requirements are met:

$$\begin{cases} \alpha_{RUL_{i,t}} \geq 1, \text{ if } RUL_{i,t} \leq RUL_w \\ 0 < \alpha_{RUL_{i,t}} < 1, \text{ if } RUL_{i,t} > RUL_w. \end{cases} \tag{5.10}$$

The scaling parameter b remains to be set. In general, $\alpha_{RUL_{i,t}}$ should increase as the RUL decreases. Therefore, $b \in \mathbb{R}^{-}$. If $b = -1$ is considered, (5.7) would present a sudden change at RUL_w, assigning a very high weight when $RUL_{i,t} = 0$ and a very low weight to all errors when $RUL_{i,t} > RUL_w$. On the other hand, $\alpha_{RUL_{i,t}}$ should present a smooth shape, gradually varying as it approaches $RUL_{i,t} = 0$. Therefore, the smooth shape is guaranteed by selecting

$$b = \frac{-1}{RUL_w}, \tag{5.11}$$

where the condition $RUL_w > 1$ must be satisfied in order to obtain the desired behavior of the weight function.

Figure. 5.2 shows the behaviors of two weight functions with different values of b for three different RUL_w values.

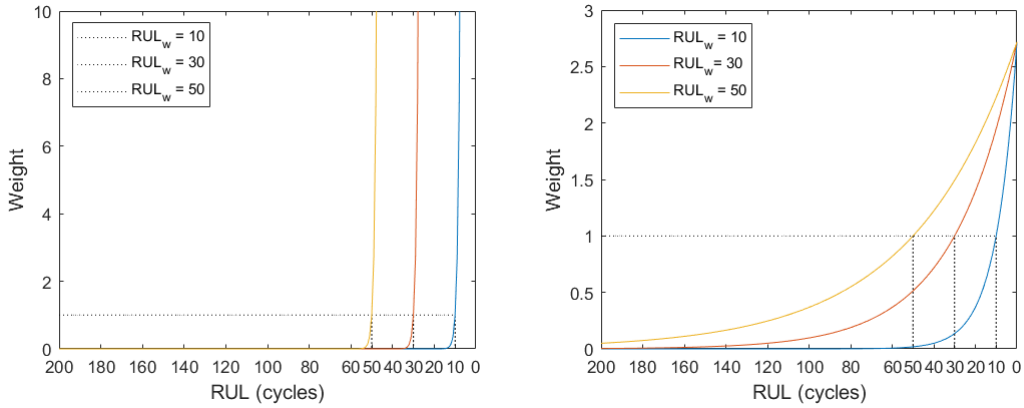


Fig. 5.2: Weight functions for $RUL_w = 10, 30, 50$: (a) $b = -1$ and (b) $b = \dfrac{-1}{RUL_w}$

Finally, the weight function obtained is

$$\alpha_{RUL_{i,t}} = \exp\left(1 - \frac{RUL_{i,t}}{RUL_w}\right). \tag{5.12}$$

Applying a weighted average approach to the prediction errors by using (5.12) as the weight function allows the $RWMSE$ metric to assess the performance of a model by taking into account the prediction accuracy during the entire degradation period. This metric allows the comparison of several prediction methods through a single value.

5.3 Procedure for RUL prediction

In this section, a procedure for RUL prediction is presented. Once a deviation in some characteristic parameter of the system has been detected, it is considered that the degradation of the system starts until

the end of its useful life. From the measurements of the system variables, it is possible to obtain a model for predicting how these variables will change with time based on the system behavior. A threshold for each variable in the system is defined to indicate the end of its useful life. The RUL of the system is determined by the time it takes for the variables reaching their corresponding predefined threshold. An advantage of the proposed degradation model is that it is adaptive. In this way, the accuracy of the RUL prediction may increase as more information is acquired. The proposed procedure is shown in Fig. 5.3. The configuration steps for this procedure are presented in Algorithm 1. A detailed explanation of each step will be given below.

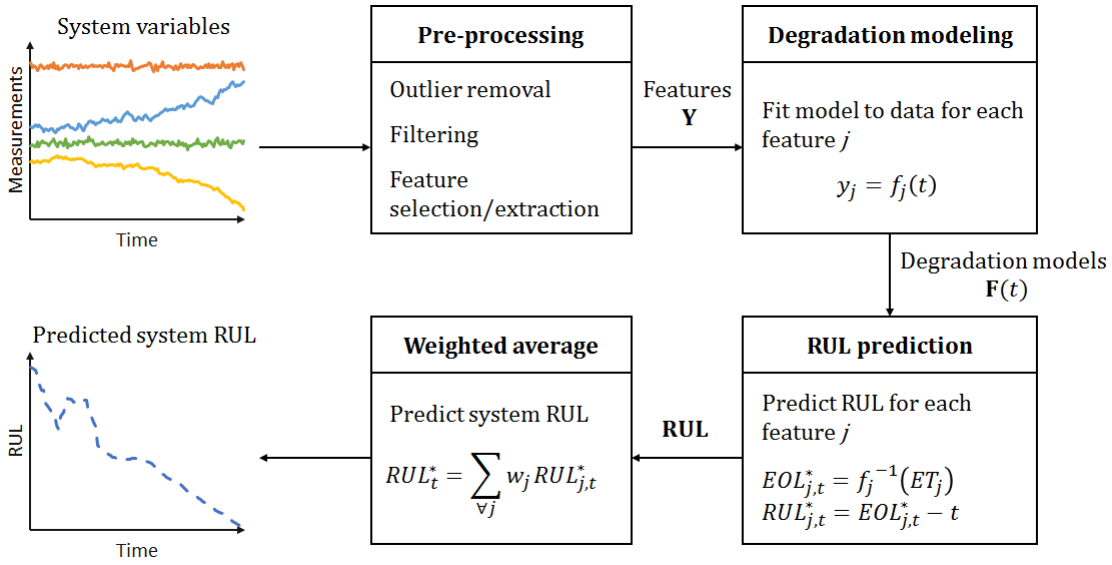


Fig. 5.3: Proposed procedure for RUL prediction

Algorithm 1 Off-line procedure

1. Apply *preprocessing* techniques to raw measurements of system variables to obtain representative features of the degradation process of the system.

for all features **do**

2. Select the *model structure* that best describes its behavior during the degradation process.

3. Define *constraints* for the selected model structure to improve degradation model prediction.

4. Obtain *EOL threshold* and *prediction model* for RUL.

end for

5. Obtain features weights for *weighted average* to predict the system RUL.

5.3.1 Preprocessing

The preprocessing stage is essential to obtain representative data of the degradation process. The data to be used in the configuration process should describe the degradation process from the moment a deviation from the nominal behavior is detected until the end of the system useful life. Preprocessing techniques are applied to extract the data relevant to the prediction model. Among the most used preprocessing techniques are:

1. Detection and removal of outliers
2. Filtering
3. Selection or extraction of features from the original variables.

In prognostics, modeling the degradation process of a system from data requires the features to reflect the progress of the degradation. If the behavior of a feature during the degradation process is stationary or constant, then it does not significantly help to predict the RUL. Therefore, features that are selected to model the degradation process should present progressive changes. Thus, metrics have been defined to select suitable prognostic features. Some of these metrics are *monotonicity*, *prognosability* and *trendability* (Coble, 2010). These metrics have been widely used to compare candidate prognostic features to determine which subset is most useful for individual-based prognostics (Coble and Hines, 2009; Coble, 2010; Ali et al., 2015; Liao et al., 2016; Saidi et al., 2017). Their values range from 0 to 1 where 0 indicates that the feature is not suitable for prognostics.

- ***Monotonicity*** quantifies monotonic trends as the system evolves toward failure (Coble, 2010). As a system gets progressively closer to failure, a suitable prognostic feature typically shows a monotonic trend. Conversely, any feature with a non-monotonic trend is less suitable. The formula to compute monotonicity is

$$Monotonicity = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{PD-ND}{T-l}\right|, \tag{5.13}$$

  where PD and ND are the number of positive and negative Δy_i differences, respectively; $\Delta y_i = y_i(t+l) - y_i(t)$ are the $T-l$ differences for each system; y_i represents the measurements of a feature of the system i; l is the number of data points used to calculate one difference; T is the number of instants of system i; and N is the number of systems.
- ***Prognosability*** is a measure of the variability of a feature at failure based on the trajectories of several run-to-failure experiments (Coble, 2010). A more prognosable feature has less variation at failure relative to the range between its initial and final values. As a system gets progressively closer to failure, a suitable prognostic feature is typically highly prognosable. Conversely, any feature that is non-prognosable is less suitable. The formula to compute prognosability is

$$Prognosability = \exp\left(-\frac{\sigma(y_i(T))}{\mu(|y_i(T)-y_i(1)|)}\right), \ \ i = 1,...,N. \tag{5.14}$$

- ***Trendability*** is a measure of similarity between the trajectories of a feature measured in several run-to-failure experiments (Coble, 2010). A more trendable feature has trajectories with the same underlying shape. As a system moves progressively closer to failure, a suitable prognostic feature is typically

highly trendable. Conversely, any feature that is non-trendable is less suitable. The formula to compute trendability is

$$Trendability = \min_{i,j} \left| corr(y_i, y_j) \right|, \;\; i, j = 1, ..., N, \tag{5.15}$$

where

$$corr(y_i, y_j) = \frac{1}{T} \frac{\sum_{t=1}^{T} (y_{i_t} - \mu(y_i))(y_{j_t} - \mu(y_j))}{\sigma(y_i)\sigma(y_j)}. \tag{5.16}$$

When y_i and y_j have different lengths, the shorter vector is resampled to match the length of the longer vector. To facilitate this process, their time vectors are first normalized to percent lifetime: 0% to 100%.

By defining a fitness function as a weighted sum of the three metrics

$$fitness = w_m Monotonicity + w_p Prognosability + w_t Trendability, \tag{5.17}$$

the features can be selected. The constants w_m, w_p and w_t weigh the importance of each metric in the selection of the features.

5.3.2 Degradation modeling

For degradation modeling, gradual degradation under invariant operational conditions is assumed, as well as the system working under invariant operational conditions during its useful life. The system degradation model is based on the deterioration of each previously selected feature, as shown in Fig. 5.3. The parameters of the degradation model corresponding to each feature are adjusted according to the behavior of the variable by applying an on-line procedure (presented in Sect. 5.3.4) and also taking into account the model structure selected from the historical data set.

5.3.2.1 Model structure

A key element in modeling the behavior of each single feature is the selection of the model structure. Possible structures can range from the simplicity of a linear model to the complexity of a deep neural network. For single variable modeling, however, the selection should always focus on the simplest possible structure.

In this work, the following model structures are considered because of their simplicity:

$$\text{Linear: } f(t, \mathbf{p}) = y = p_1 t + p_2 \tag{5.18}$$

$$\text{Quadratic: } f(t, \mathbf{p}) = y = p_1 t^2 + p_2 t + p_3 \tag{5.19}$$

$$\text{Exponential: } f(t, \mathbf{p}) = y = p_1 e^{p_2 t} + p_3, \tag{5.20}$$

where y and t are the dependent and independent terms, respectively, and $\mathbf{p} = [p_1, p_2, p_3]$ is the vector of parameters of f to be fitted. As a measure of the error made in estimating a variable during the degradation process, the root of the mean squared error (RMSE) is used:

$$RMSE(y, \mathbf{p}) = \sqrt{\frac{\sum_{t=t_d}^{T} (f(t, \mathbf{p}) - y_t)^2}{T}}, \tag{5.21}$$

where t_d is the detection time of the degradation process, y_t is the expected value at instant t, and T is the total number of observations. Figure 5.4 shows an example of the response of the models obtained when fitting these structures to the data of an artificial feature.

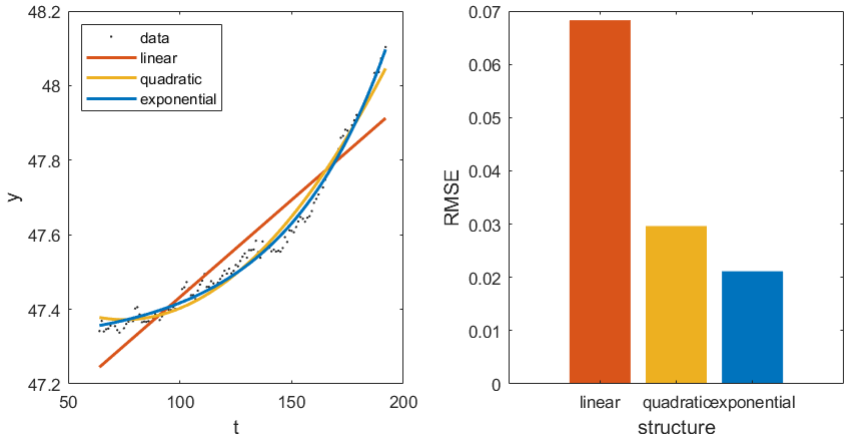


Fig. 5.4: Fit of different structures to the data of a variable

The structure for modeling each feature is selected independently since each feature may have a different evolution during the degradation of the system. Previous studies, which mainly focused on the construction of system health indices, present the exponential model on a recurring basis as part of the probabilistic models used to describe the health of systems that have cumulative degradation (Ali et al., 2018; Saidi et al., 2017; Dong et al., 2019).

5.3.2.2 Model constraints

Once a structure is selected, its ability to predict the future behavior of the degradation process is evaluated. While fitting a degradation model, both the fitting error of previous measurements and the prediction error of future measurements are considered in this chapter. If the prediction error increases, it would be necessary to constraint the parameter space of the model to reduce it. With the addition of regularization constraints to the fitting function, a search region is generated where two requirements are satisfied:

1. Minimize the model fit error for adapting to the current behavior of the feature during the degradation process and

2. Remove parameter values that introduce large errors in predicting future degradation behavior.

Fitting of the degradation model on-line is therefore presented as the following constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{p}} \ RMSE(y, \mathbf{p}) \\ &\text{s.t. } \mathbf{p} \in R_s, \end{aligned} \tag{5.22}$$

where y are the values of the features that represent the process degradation, R_s is the allowed region for parameter values of the selected structure. R_s is defined based on historical data of similar systems.

The simplest constraint defines constant lower and upper limits for each parameter p_i, creating what is called a box constraint. The type of constraint, however, may vary depending on the structure defined to model the degradation process. As the exponential structure is selected in this chapter to model the degradation process of each single feature, it is necessary to delve into the analysis of constraints for this type of structure.

Constraints for exponential structure

When analyzing the on-line response of the exponential degradation model, from the beginning of the degradation process to the end of the system's useful life, it is possible to see a wide range of model behaviors. Figure 5.5 shows how a model fits and predicts the behavior of an artificial variable at three instants during the degradation process ($t = [94, 124, 164]$).

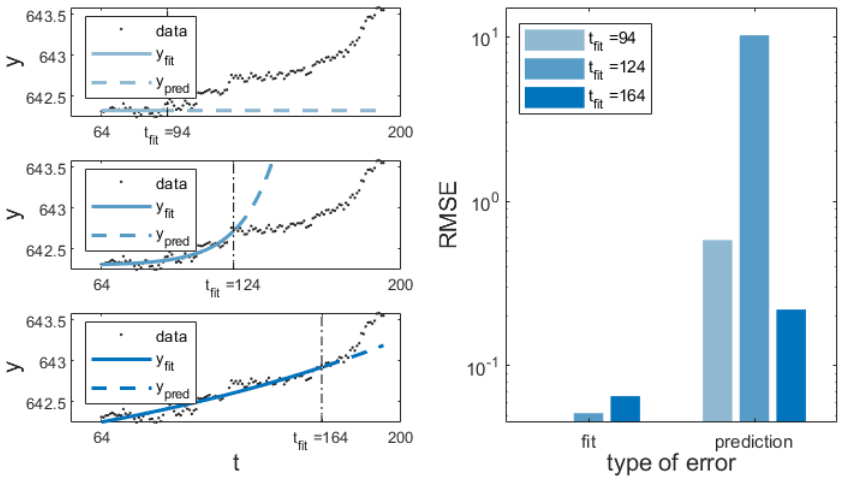


Fig. 5.5: On-line fitting of an exponential model to the data of one variable

The need for a constrained parameter space when the model of the feature is adjusted on-line will be explained based on Fig. 5.5. The fitting error for the feature shown validates the selection of the exponential structure. The error made in the prediction, however, is high, mainly at the beginning of the degradation process when there are few measurements of the variable. This is because the standard

performance measure (5.21) only focuses on minimizing the fitting error. At the start of the degradation process, the range of possible parameter values for which a good fit of the model is obtained is wide. A small fitting error for the initial data does not guarantee a small prediction error, as is the case of the model response for $t = 94$ and $t = 124$. The model response shows a constant behavior of the variable for $t = 94$, indicating that the system is not degrading. When the model is updated at time $t = 124$, the figure shows how the variable changes in response to the degradation process. The rate of degradation, however, is much higher than the true one. Both responses therefore describe behaviors of the variable that are not characteristic of the degradation process. When the acquired measurements of the variable reflect the entire progress of the degradation of the system, the range of possible values of the parameters decreases, and its response presents a better prediction. The primary objective of constraint enforcement is therefore to decrease the range of possible values for the degradation model parameters when a small amount of data is available.

The steps for configuring the constraints of the exponential structure (parameters p_1, p_2 and p_3 of Eq. (5.20)) are described below.

1. *Characterize the historical behavior of the feature.* It is necessary to adjust the parameters of the degradation model in such a way that its response adjusts as much as possible to the behavior of the feature through the entire degradation process.
   Since the exponential structure is nonlinear, it is proposed to use the trust-region reflective method (Coleman and Li, 1996b,a; Branch et al., 1999). Its main advantage is the speed of convergence. The convergence depends, however, on the search region of the parameters. Therefore, it is proposed to use the three-sigma procedure, presented in Algorithm 2, to reduce this region, starting from defining upper and lower limits for the parameters and to obtain the values of the parameters of the degradation model that characterize the historical behavior of the feature during the degradation process.
2. *Define the constraint.* By applying the procedure three-sigma setting $tolerance = 10^{-5}$, a first box constraint was obtained for the optimization problem stated in (5.22):

$$R_s = \{\mathbf{p} \in \Re^3 : p_{kL} \leq p_k \leq p_{kU}, k = 1,2,3\}. \tag{5.23}$$

   Once a constraint is defined, the fitting and prediction errors of the degradation model are evaluated.
   To define the type of constraint for each parameter, it is necessary to analyze the relationship between the parameters of the exponential structure (p_1, p_2 and p_3) and the behavior of the feature. Figure 5.6 shows the influence of each parameter on the exponential model. Parameters p_1 and p_2 determine the curvature of the model, so they are related to the rate of change of the feature. Similar systems can present different degrees of degradation, resulting in degradation models with different values of p_1 and p_2. The parameter p_3 represents the position of the asymptote on the y-axis. Values of the feature must be above or below this value, depending on the feature. Considering that the behavior of the feature is stationary during the normal operation of the system, it is possible to estimate the parameter p_3 from measurements acquired before the start of the degradation process. Estimating the value of the parameter p_3 from the start of the degradation process incorporates stability to the process of fitting the degradation model and, consequently, to the prediction of the RUL.
3. *Estimation of* p_3. The value of p_3 is estimated from the mean (μ) and the standard deviation (σ) of the measurements before the start of the degradation process. Since p_3 must function as an asymptote, three possible estimators were considered:

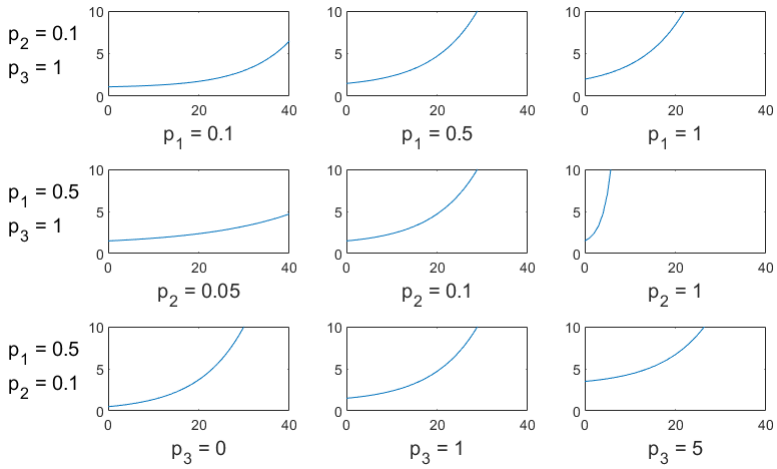


Fig. 5.6: Influence of the parameters p_1, p_2 and p_3 on the behavior of the model. Two parameters are held fixed in each row and the third parameter is varied

$$\mu + \sigma \text{ or } \mu - \sigma \tag{5.24}$$
$$\mu + 2\sigma \text{ or } \mu - 2\sigma \tag{5.25}$$
$$\mu + 3\sigma \text{ or } \mu - 3\sigma. \tag{5.26}$$

Algorithm 2 Procedure three-sigma to set the initial values of the degradation model parameters

1. Initialize the search region based on the relationship between parameters (p_k, $k = 1,2,3$) and feature measurements (y). To solve the optimization problem (5.22), the following constraint is considered:

$$\begin{aligned}(p_{1L_0}, p_{1U_0}) &= \begin{cases}(0,\text{inf}), & \text{if } y(T) - y(1) > 0\\ (-\text{inf},0), & \text{if } y(T) - y(1) < 0\end{cases}\\ (p_{2L_0}, p_{2U_0}) &= (0,\text{inf})\\ (p_{3L_0}, p_{3U_0}) &= (\min(y), \max(y))\\ R_s &= \{p_{kL_0} \le p_k \le p_{kU_0}\}.\end{aligned}$$

repeat

2. Obtain sets of parameters P_k by solving Eq.(5.22) considering R_s.

3. Create new limits:

$$\begin{aligned}p_{kL_{new}} &= \mu_{P_k} - 3\sigma_{P_k}\\ p_{kU_{new}} &= \mu_{P_k} + 3\sigma_{P_k}\\ R_s &= \{p_{kL_{new}} \le p_k \le p_{kU_{new}}\},\end{aligned}$$

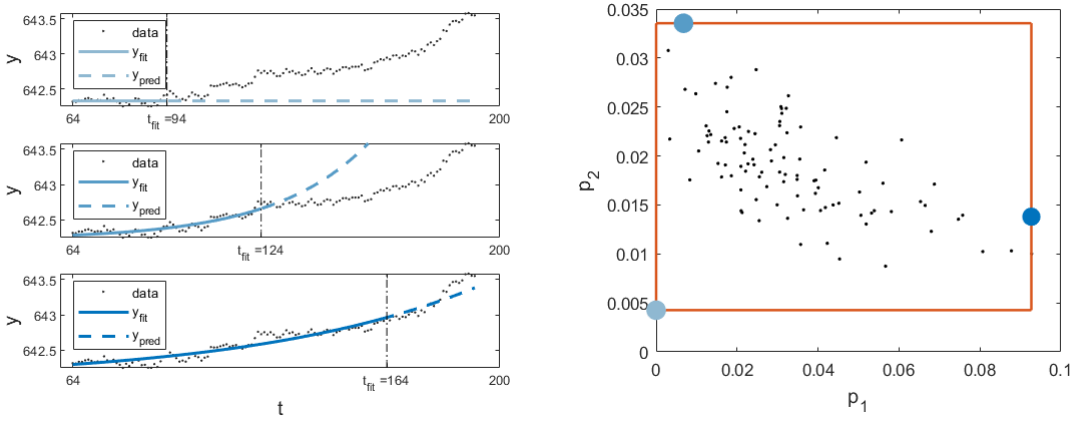
where μ_{P_k} and σ_{P_k} are the mean and standard deviation of the values of the set P_k.

until

$$\left|p_{kL_{new}} - p_{kL_{previous}}\right| < tolerance \quad \text{and} \quad \left|p_{kU_{new}} - p_{kU_{previous}}\right| < tolerance$$

according to the monotony of the feature during the degradation process. The estimation of the parameter p_3 from the beginning of the degradation process also allows reducing the complexity of the model.

4. *Constraint for p_1 and p_2.* From the values of the parameters of the degradation model for various systems, information is available about different degradation paths of similar systems, which is reflected in the values of the parameters p_1 and p_2. With the values of each parameter, a first approach is to define lower and upper limits for each parameter, creating what is called a box constraint. Figure5.7(a) shows the model response of an artificial variable obtained by fitting the parameters in three different instants of the degradation of a system. At the beginning of degradation process, when measurements contain little information, the model responses obtained show future degradation paths that differ significantly from those recorded.
   If the different values of the degradation model parameters obtained for various systems are plotted, in a two-dimensional space where each dimension corresponds to a parameter, as shown in Figure 5.7(b), the distribution of the parameters does not correspond to the box constraints defined above. The figure highlights three points that correspond to the values of p_1 and p_2 that generate the model responses shown in Figure 5.7(a). If the distribution of these points is analyzed, it is clear that the point corresponding to the model update at $t = 94$ is located in a corner of the box. The cluster of points shown in Figure 5.7(b) is considered the space that contains the values of the parameters p_1 and p_2 that characterize the degradation process of these systems. Any value of p_1 and p_2 generated by a point outside this cloud generates a model response that does not accurately represent the degradation process. Therefore, the constraint for p_1 and p_2 should be modeled using a region that conforms to the way the values are arranged in this space.



(a) (b)

Fig. 5.7: Fitting the exponential model with a box constraint

A model based on support vector data description (SVDD) is proposed to model the constraint for the parameters p_1 and p_2. SVDD allows creating a non-linear decision surface around a data set (Tax and Dui, 2004). Given a set of training data $\{\mathbf{x}_i\} \in \mathbb{R}^n$, $i = 1, ..., l$, the following optimization problem is solved:

$$\begin{aligned} &\min_{R,\vec{q},\vec{\xi}} \; R^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{s.t. } \|\phi(\mathbf{x}_i)-\mathbf{q}\|^2 \leq (R^2+\xi_i),\;\; i=1,...,l \\ &\qquad \xi_i \geq 0,\;\; i=1,...,l, \end{aligned} \tag{5.27}$$

where $\phi(\mathbf{x})$ is a function mapping data to a higher dimensional space, $\xi = \{\xi_1,...,\xi_l\}$, ξ_i is the slack variable of i-th training sample, and $C > 0$ is a user-defined parameter. After (5.27) is solved, a hyperspherical model is characterized by a center $\mathbf{q}$ and a radius R. A testing instance $\mathbf{x}$ is detected as an outlier if

$$\|\phi(\mathbf{x})-\mathbf{q}\|^2 > R^2. \tag{5.28}$$

From the values of the parameters p_1 and p_2, a SVDD model characterizes its distribution and, therefore, the behavior of the feature during the degradation process. The SVDD model does not, however, guarantee that all points are within the hypersphere. Therefore, a threshold r_{th} is added to R so that all the degradation patterns are considered within the search region.
Taking into account the previous analysis, the constraint of the parameters p_1 and p_2 is defined by

$$R_s = \{\|\phi(\mathbf{p})-\mathbf{q}\| \leq R + r_{th}\}. \tag{5.29}$$

Figure 5.8(a) shows the degradation model responses obtained by fitting the parameters in three instants of the degradation of a system, using the constraint defined in (5.29). The system corresponds to that represented in Fig. 5.7(a). Figure 5.8(b) represents the SVDD model that defines the constraint, in which the points corresponding to the values of the parameters p_1 and p_2 are highlighted for the model responses shown in Fig. 5.8(a). The constraint allows bringing the model's response closer to the real behavior of the feature because as the model is updated by decreasing the fitting error to the acquired measurements, it is guaranteed that the values of the parameters p_1 and p_2 are within the region defined by the SVDD model.

5.3.3 RUL prediction

In this section, the method for predicting the RUL and the degradation model defined in the previous section is presented.

5.3.3.1 EOL threshold

The EOL threshold (ET), or failure threshold as it is also known, is a value of the feature at which point components or subsystems can be repaired or replaced immediately before the critical failure occurs (Bregon and Daigle, 2019). In most cases, the ET is estimated based on the physical knowledge of the system (Lei et al., 2016; Zhang et al., 2017; Yan et al., 2020). Expert knowledge, however, is not always accessible, and this value is often difficult to determine. To address this issue, some works consider

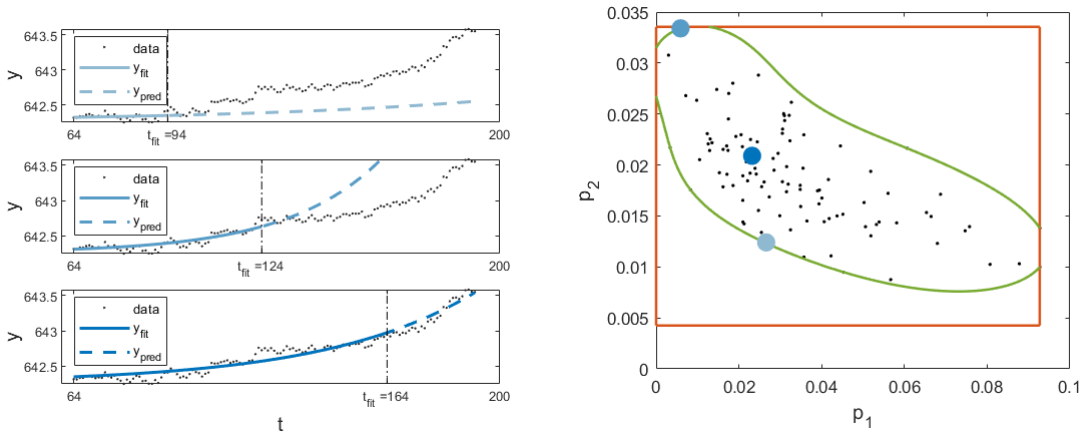


Fig. 5.8: Fitting the exponential model with the SVDD constraint

statistics like the mean of the last measurements of multiple systems as an estimation of the failure threshold (Liu et al., 2013). Since guaranteeing that these measurements follow a normal distribution for every type of system is difficult, the mean is not always an adequate estimator. In this chapter, the ET is estimated from the inverse cumulative distribution function of these measurements, with the assumption that they follow a unimodal distribution.

From the historical data of N systems, the last measurement of each variable y_T is obtained for each system. The ET is determined as the value with probability $1-p$ that the feature reaches the threshold during the system degradation:

$$ET = CDF^{-1}(p), 0 \leq p \leq 1, \tag{5.30}$$

where CDF^{-1} is the inverse cumulative distribution function of $y_{i,T}$, $i = 1,...,N$.

In the feature threshold configuration, several values of p must be considered according to the characteristics of the data. To determine which value of p is best to obtain ET, the RMSE is used as a measure of the prediction error of the EOL of the systems in the historical data:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (EOL^*_{i,T} - EOL_i)^2}{N}}, \tag{5.31}$$

where $EOL^*_{i,T}$ is the predicted EOL of the system i at the last moment of the useful life T, EOL_i is the true EOL of the system i, and N is the number of systems. The value of EOL_i is determined by the duration of the useful life of the system. The value of p selected to determine the threshold has the lowest prediction error.

In this work, the following definition for RUL included in the International Organization for Standardization (ISO) 13381-1:2015 is used; see Definition 5.2.

Definition 5.2. *Remaining useful life* . Remaining time before system's health falls below a defined failure threshold (ISO, 2015).

Therefore, once the EOL has been predicted, the RUL is determined according to (5.32) with t being the time at which the prediction is made. The periodicity at which the RUL is predicted must be determined, according to the criteria of experts.

Definition 5.3. *Remaining useful life of a system considering a single feature*. The remaining useful life of a system at continuous time t depending on a single feature y_j is

$$RUL^*_{j,t} = EOL^*_{j,t} - t \tag{5.32}$$

where

$$EOL^*_{j,t} = inf\{t_k \in \Re : t_k > t \ \wedge \ f_j^{-1}(t_k) = ET_j\}, \tag{5.33}$$

with ET_j being a performance threshold.

5.3.3.2 Prediction model

The prediction of the EOL at each moment during the process degradation can be performed by evaluating the degradation model in future instants until the estimated value exceeds the predefined threshold. This operation, however, is computationally expensive and may introduce errors in the prediction depending on the resolution of the degradation model evaluation. Conversely, if the inverse of the degradation model is obtained, the EOL can be predicted by evaluating the threshold.

By taking into account that the degradation process is modeled using an exponential structure and by evaluating (5.20) for the EOL prediction, $(t,y) = (EOL^*_{j,t}, ET_j)$, the following expression for ET_j is obtained:

$$ET_j = p_1 \exp(p_2 EOL^*_{j,t}) + p_3. \tag{5.34}$$

Thus, the prediction model is obtained by the inverse of the model represented in (5.34)

$$EOL^*_{j,t} = \frac{\ln(\frac{ET_j - p_3}{p_1})}{p_2}. \tag{5.35}$$

The parameters p_1, p_2 and p_3 correspond to those estimated for the degradation model. Figure 5.9 shows the response of the prediction model that allows checking the inverse behavior of the degradation model and visualizing the direct prediction of the EOL.

The explained procedure for the degradation modeling and the prediction of the EOL and the RUL is implemented for each feature.

5.3.3.3 Weighted average

The prediction model for each feature provides information on the progress of the degradation according to its effect on each one of them. The prediction of the RUL of the system should therefore reflect the contribution of each feature. The RUL of the system is then determined by the weighted average of the predicted RUL for each feature as established in (5.36).

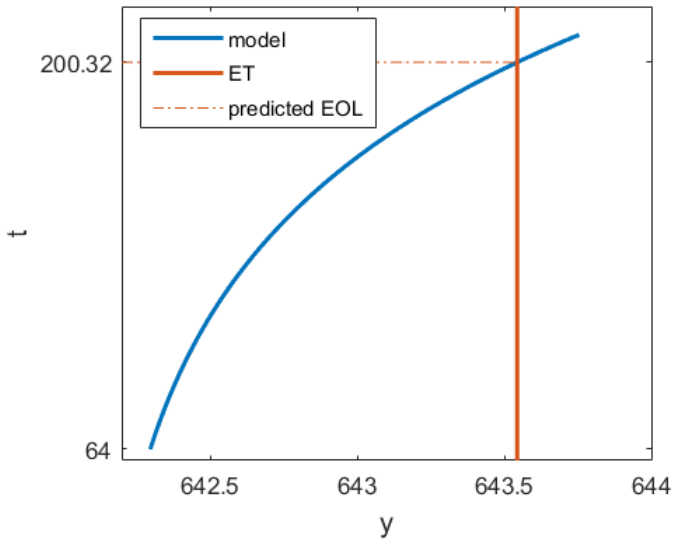


Fig. 5.9: Prediction model response

Definition 5.4. *Remaining useful life of a system considering a set of features*. The remaining useful life of a system i at continuous time t depending on a set of V features is

$$RUL^*_{i,t} = \frac{\sum_{j=1}^{V} w_j RUL^*_{j,t}}{V} \tag{5.36}$$

where $RUL^*_{j,i,t}$ is the RUL given feature j at time t and w_j is a weight scalar assigned to feature j according to its contribution in the prediction of the system's RUL.

Therefore, the next step in the configuration is to determine the contribution of each feature. The adjustment of the weights is presented as an optimization problem, where the objective is to minimize the RUL prediction error. The proposed performance metric, defined in (5.5), is used to calculate the RUL prediction error. The optimization problem is defined as

$$\begin{aligned} &\min_{w_j} \; RWMSE(RUL^*_{j,t}, RUL_{i,t}, w_j) \\ &\text{s.t.} \quad \sum_{j=1}^{V} w_j = 1 \\ &\qquad\quad w_j \leq 1, \forall j. \end{aligned} \tag{5.37}$$

In case of removing features from the weighted average, it is necessary to solve the optimization problem in (5.37) again in order to satisfy its constraints.

5.3.4 On-line procedure

After a configuration process, the prognostic procedure can be implemented to predict the RUL of a system. The on-line procedure for predicting the RUL of a system, presented in Algorithm 3, is implemented through a cumulative window approach. This approach requires the addition of two parameters: *minsize* or *minimum size* of the window of observations used to fit the degradation model, and *stride* or sampling time considered for updating the degradation model. As measurements are acquired, model parameters are recalculated to adapt to the current state of system degradation.

Algorithm 3 On-line procedure for RUL prediction

Parameters: *minsize* signifies *minimum size* of the window of observations required to estimate the parameters of a degradation model, and *stride* means sampling time considered for updating the degradation model

Inputs: t_d is the degradation detection time, R_s is the constrained region of the parameter space, and $\mathbf{w}$ is the weight vector that considers the importance of each feature for the accurate prediction of the system RUL

Initialization: $\mathbf{t}_{window} \leftarrow t_d$, $\mathbf{M}_{window} \leftarrow \emptyset$, $nobs \leftarrow 0$
repeat
  Acquire measurements $\mathbf{m}_t$ at time t
  $\mathbf{t}_{window} \leftarrow \mathbf{t}_{window} \bigcup t$
  $\mathbf{M}_{window} \leftarrow \mathbf{M}_{window} \bigcup \mathbf{m}_t$
  $nobs \leftarrow nobs + 1$
  if $nobs > minsize$ **then**
    Preprocess $\mathbf{M}_{window}$ to obtain features $\mathbf{Y}$
    for all $\mathbf{y}_j \in \mathbf{Y}$ **do**
      Obtain f_j given $\mathbf{M}_{window}$ by solving Eq. (5.22) considering R_s
      Predict EOL at current time $EOL^*_{j,i,t}$ for feature j
      $RUL^*_{j,t} \leftarrow EOL^*_{j,t} - t$
    end for
    Predict system RUL through Eq. (5.36) considering $\mathbf{w}$.
  end if
  $t \leftarrow t + stride$
until system failure

5.4 Application of the proposal

In this section, the application of the proposal to the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) is presented (Saxena et al., 2008). Furthermore, the proposed model is compared with a state-of-the-art DNN-based model.

5.4.1 Case study

The C-MAPSS prognostics' case study is formed by four distinct data sets. Each one contains measurements of 21 sensors and three operating conditions. Each data set considers degradation information of several engines with both training and testing examples. To validate the proposal, the first data set containing information about one operating condition (altitude 0 ft, throttle angle 100 deg and Mach 0) and one type of failure (HPC degradation) is used. The data set contains a number of training engines (100 engines: training (N)) with run-to-failure information and a number of testing engines (100 engines: testing) with information terminating before a failure is observed.

Engines start operating with different degrees of initial wear but are considered healthy. As the number of cycles increases, each engine starts to deteriorate until failure. The training data set contains run-to-failure information for each engine. The testing data set contains temporal data up to a certain time before the engine fails. The objective is to predict the RUL of testing engines (Saxena et al., 2008).

In the experiments considered here, the training set is used for both training and testing/validating the procedure because the testing data set does not contain labeled information that can be used to verify the proposed procedure's performance.

5.4.2 Preprocessing

The training data set considered (FD001) is contaminated with noise, which makes analyzing the degradation trend difficult. The presence of noise in the measurements hinders the RUL prediction; thus filtering is applied as explained below.

Filtering. EWMA (exponential weighted moving average) filter is applied for removing noise. The expression that describes the filter is

$$\begin{aligned} y_0 &= x_0 \\ y_i &= \alpha x_i + (1-\alpha) y_{i-1} \end{aligned}. \tag{5.38}$$

where x_i represents the original data vector, y_i the filtered data vector, and α is the exponential percentage, defined as $\frac{2}{l+1}$, where l is the size of the window of observations to be filtered. For all variables, $l = 15$ is used.

Feature selection. Feature selection is performed by evaluating the fitness function in (5.17) for each variable and rejecting the variables with fitness below a predefined threshold. Monotonicity calculations for all variables use $l = 30$. The constants w_m, w_p and w_t for this application are set to 1 weighting equally for each metric. A threshold equal to 1.5 is selected since a sum is used as fitness function (Coble, 2010).

Figure 5.10 shows each metric and the fitness score for each variable. Sensors 1, 5, 6, 9, 10, 14, 16, 17, 18, and 19 have a fitness score under 1.5, which means that they are not very useful for prognostics. Therefore, these variables are not selected for modeling the degradation process.

Figure 5.11 shows the behavior of the variables used for modeling the degradation process for engine 1. These variables correspond to sensors 2, 3, 4, 7, 8, 11, 12, 13, 15, 20, and 21.

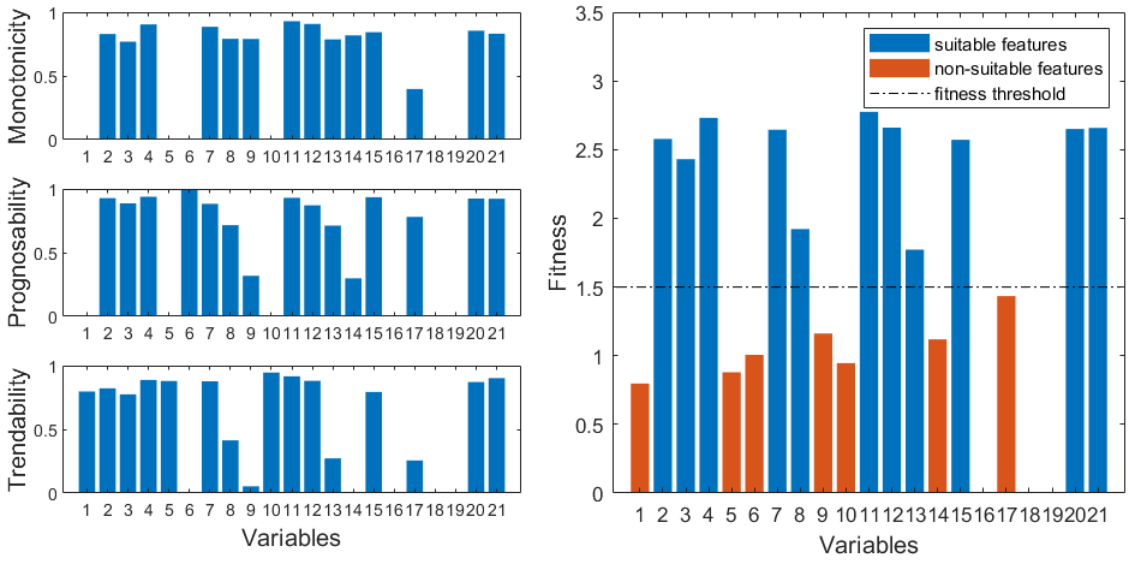


Fig. 5.10: Results of the feature selection process

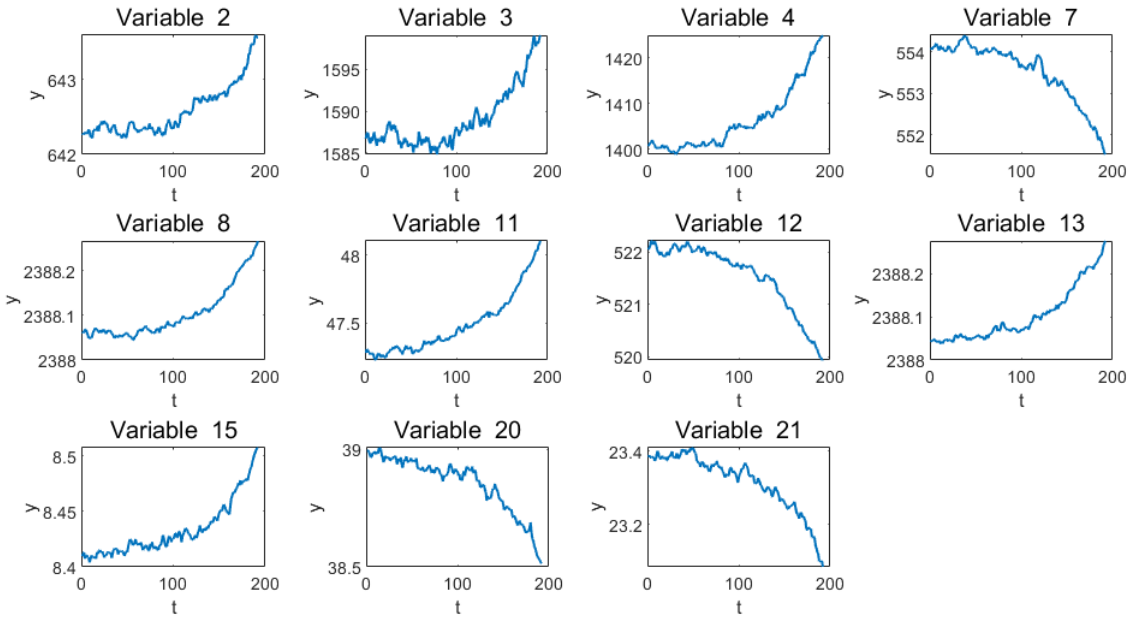


Fig. 5.11: Behavior of the selected variables to model the degradation process for engine 1.

5.4.3 Model training

Building each prediction model consists of selecting its structure as well as its constraints, establishing the EOL threshold, and estimating the weight of the variables for predicting the RUL of each engine.

5.4.3.1 Model structure

From the behavior of the variables selected in the preprocessing stage, the quadratic and exponential structures are considered for modeling the degradation process. Figure 5.12 shows the fitting error of both structures for the selected variables. The procedure presented in Sect. 5.3.2.2 is applied to fit the exponential structure with box constraints. The fitting error is averaged for each variable of the 100 engines of training data. The exponential structure is finally selected for modeling the degradation process because of its lower fitting error compared with the quadratic structure.

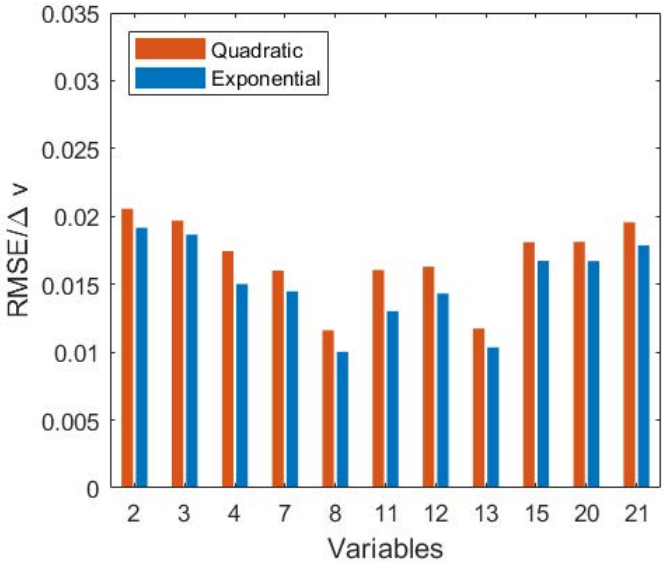


Fig. 5.12: Fitting error of the degradation models for each variable with quadratic and exponential structures. The error has been normalized with respect to the magnitude of each variable.

5.4.3.2 Model constraints

Estimation of p_3. For each estimator presented in (5.24), (5.25), and (5.26), the exponential structure prediction model of each variable was obtained. The fitting error of the degradation models (Fig. 5.13) shows that the best estimator for the parameter p_3 is $\mu + \sigma$ or $\mu - \sigma$, depending on the monotony of the variable.

Constraints for p_1 *and* p_2. The parameters obtained from the models in the estimation of p_3 are used to fit a SVDD model for each variable as a constraint for the parameters p_1 and p_2. In the SVDD model, the Gaussian kernel is used as the mapping function ϕ (5.39), with a parameter h that represents the bandwidth of the Gaussian function.

$$\phi(p_1, p_2) = \exp\left(-\frac{\|\langle p_1, p_2\rangle\|^2}{h^2}\right). \tag{5.39}$$

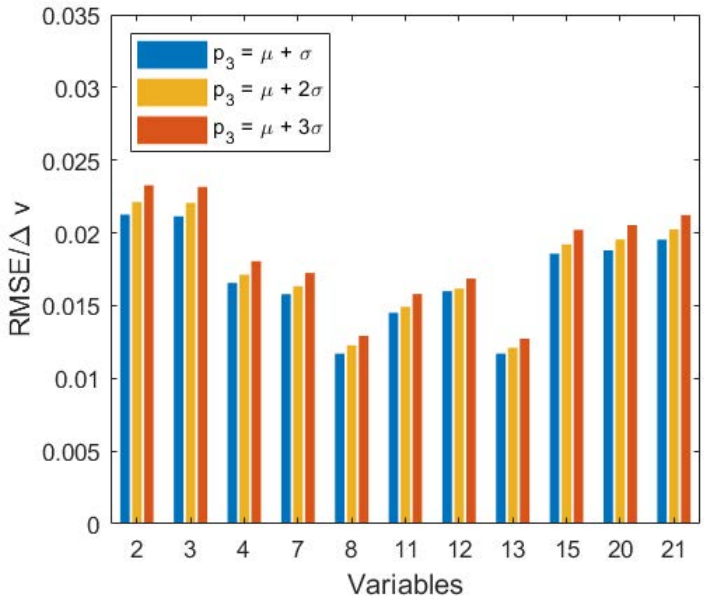


Fig. 5.13: Fitting error for the degradation models with different estimators of p_3. The error has been normalized with respect to the magnitude of each variable.

The distribution of the parameters for each variable and the SVDD model obtained are shown in Fig. 5.14. C and h in (5.27) and (5.39) for each SVDD model shown in Table 5.3. These parameters are manually tuned such that all models remain within the decision surface, excluding the space that does not contain previously fitted degradation models. Subsequently, the degradation models are readjusted to validate the defined constraints (Fig. 5.15). The fitting error obtained with the new constraints may vary with respect to the error obtained with the box constraints. It is important to emphasize that the acceptable fitting error depends on the user. In that case, tuning the constraints can be accepted in order to decrease the on-line prediction error.

Table 5.3: Values of the parameters of the SVDD model for each variable

Variable	2	3	4	7	8	11	12	13	15	20	21
C	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
h	2.3	2.7	2.4	2.3	2.5	2.3	2.3	2.3	2.6	2.3	2.2

5.4.3.3 EOL threshold

To select the threshold for each variable, the EOL prediction error is calculated for several values of p, by using all degradation measurements. The values of p are 0.1, 0.3, and 0.5 since these are values with which the probability of reaching the threshold is greater than or equal to 0.5. To this end, the EOL

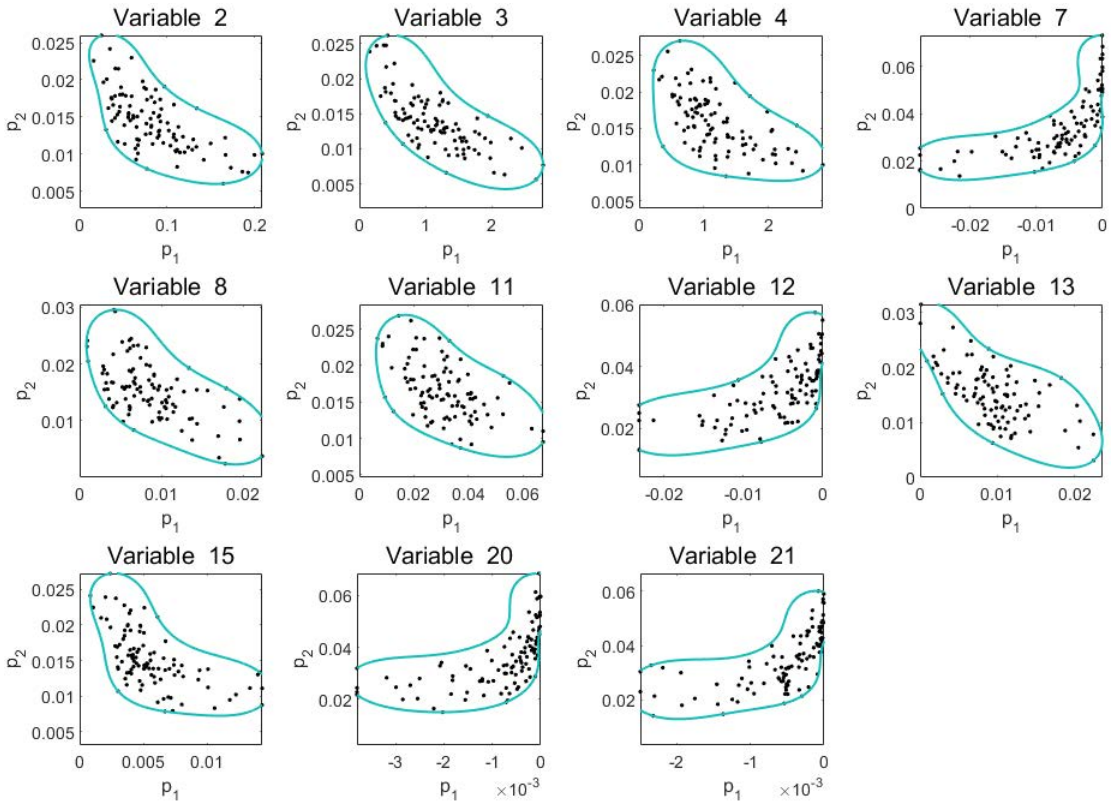


Fig. 5.14: Constraints of p_1 and p_2 for each variable.

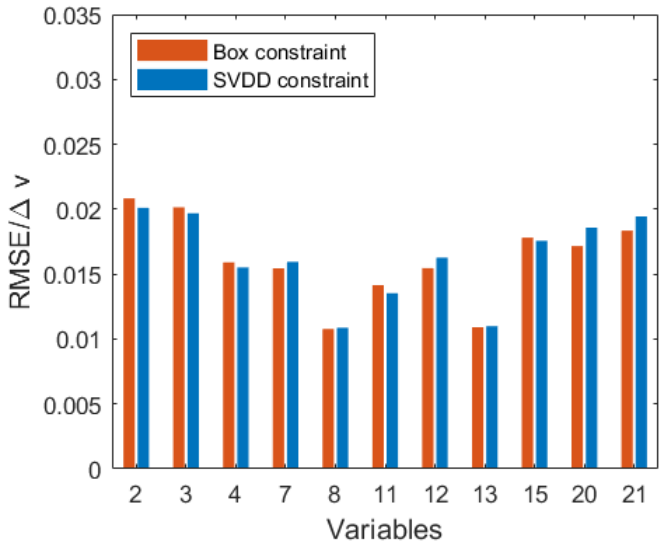


Fig. 5.15: Fitting error of the degradation models for each variable with box constraints and SVDD. The error has been normalized with respect to the magnitude of each variable

estimation error of the engines of the training set is averaged for each variable. From the results shown in Fig. 5.16, the criterion for determining the threshold for each variable is the value for $p = 0.5$.

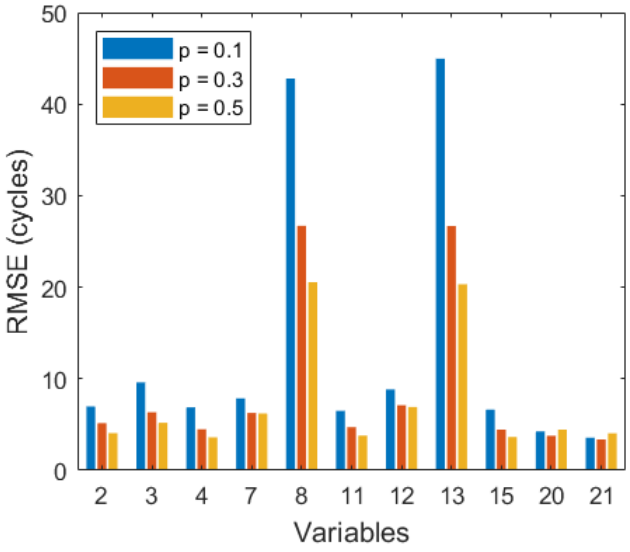


Fig. 5.16: EOL prediction error at the time of failure

5.4.3.4 Features weights

Adjusting the features weights requires evaluating the RUL prediction error on-line. To do this, the degradation models are fitted, and the RUL is predicted on-line for each variable of the 100 engines in the training set. A first step, which is recommended for the comparison, is to equitably assign weights to the variables and calculate the RUL prediction error. This value will serve as a reference for solving the optimization problem presented in (5.37), which allows obtaining the weights that minimize the RUL prediction error. Figure 5.17 represents the effect of the weight of the variables on the RUL prediction of the system by comparing the prediction error before and after the optimization of the weights. It can be observed that by optimizing the weights the combination of variables obtained significantly reduces the prediction error of the RUL.

From the weights presented in Fig. 5.17, the contribution of each variable in the prediction of the RUL can be determined. Thus, it is evident that variables 8 and 13 do not contribute significantly. These variables can be removed from the prognostics model, and, in this case, the weight of the remaining variables must be optimized again.

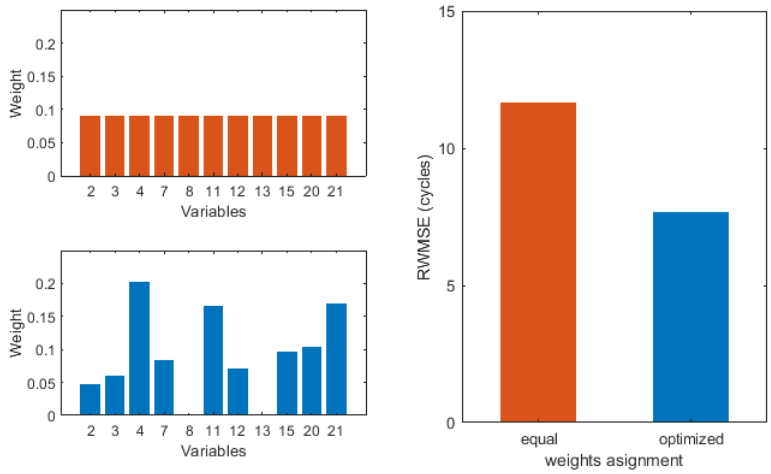


Fig. 5.17: Influence of the weight of the variables on the prediction of the RUL of the system

5.4.4 Model validation

During the construction of the prediction model, it is possible to evaluate each step using the training data. To validate the model, however, it is necessary to present data on the complete degradation process of engines not used in training. Therefore, the k-fold cross-validation method is used to validate the prediction model on the data of the 100 engines of the training set. For cross-validation, $k = 10$ is used, while also employing in each partition 90 training engines and 10 testing engines.

The proposed method for estimating RUL is compared with a DNN model (Song et al., 2018). This prediction model consists of the combination of two neural network structures: an autoencoder (AE) and a bidirectional long short-term memory neural network (BLSTM). By assuming the values presented in the chapter, the prediction model was implemented and the cross-validation method was applied. The same partitions were used for both prediction models.

For each partition, the off-line procedure presented in Algorithm 1 is used for the construction of the prediction model with the 90 engines in the training set. Algorithm 3 is implemented for the on-line RUL prediction of the 10 engines in the testing set. The *minimum size* for the cumulative window is set to 30, and the *stride* is set to 1, in order to compare the results with the DNN model.

The prediction error is calculated using the metric proposed in (5.5) with the value $RUL_w = [30, 40, 50]$. The prediction error for each partition is shown in Table 5.4.

The prediction errors of each model in each partition are compared by using the Wilcoxon statistical test. The significance level (α) of the test is 0.05. Table 5.5 shows the $p - value$ obtained when applying the statistical test. The test results, $p - value < \alpha$, show a significant difference between the prediction errors of both models, in which the proposed model in this chapter presents the smaller error.

The results shown in Fig. 5.18 demonstrate that the proposed model has better accuracy in the prediction of the RUL as the engines approach the end of their useful life.

Figure 5.19 shows the on-line fitting of the degradation model and the RUL prediction for variable 4 at three instants $t = [102, 132, 162]$ during the degradation process of engine 1. At each instant, the degradation model is updated with the acquired measurements, adapting to the engine conditions, and

Table 5.4: RWMSE. RUL prediction error of the proposed prediction model and the DNN-based model for each partition

	$RUL_w = 30$		$RUL_w = 40$		$RUL_w = 50$	
k	Proposal	BLSTM	Proposal	BLSTM	Proposal	BLSTM
1	8.81	10.51	10.11	12.32	10.92	12.97
2	8.56	10.30	10.54	12.44	12.01	13.32
3	7.36	8.59	8.87	10.56	9.98	11.43
4	8.81	12.66	10.30	14.75	11.30	15.92
5	7.45	10.51	8.97	12.60	10.10	13.66
6	8.92	9.56	10.93	11.68	12.41	12.91
7	10.52	10.25	12.99	12.36	14.83	13.67
8	7.14	10.66	7.98	12.08	8.56	12.83
9	10.27	12.27	11.58	15.16	12.75	16.52
10	6.67	14.21	7.80	15.85	8.59	16.52
average	**8.45**	10.95	**10.00**	12.98	**11.15**	13.98

Table 5.5: Results of applying the Wilcoxon test with $\alpha = 0.5$

	$RUL_w = 30$	$RUL_w = 40$	$RUL_w = 50$
$p - value$	1.16e-04	1.47e-04	6.92e-04

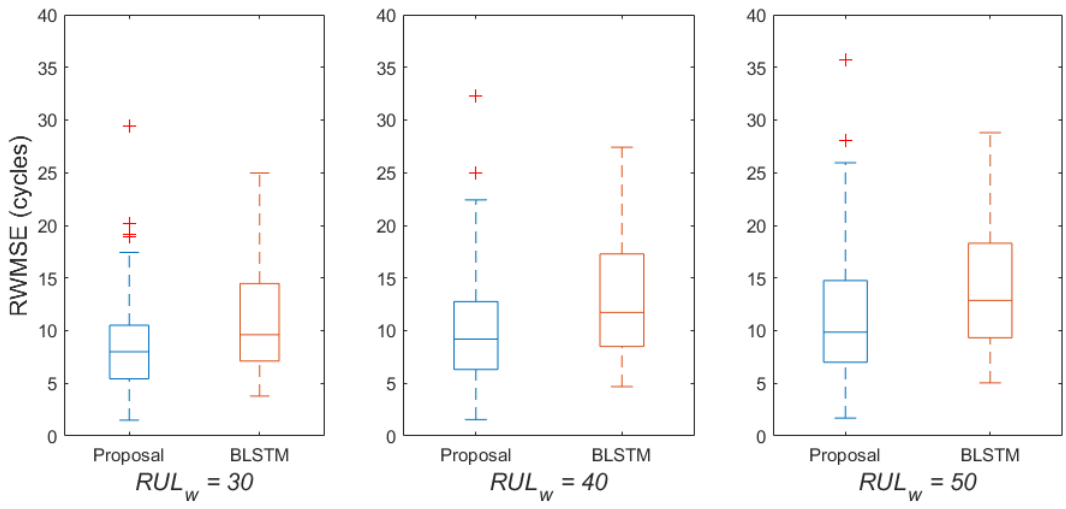


Fig. 5.18: Comparison of the RUL prediction error between the proposal and the DNN

progressively improving the RUL prediction. The predicted RUL for engine 1 throughout the degradation process, obtained from the weighted average of the predicted RUL for each variable, is presented in Fig. 5.20. The figure also presents the predicted RUL for engine 1 with the BLSTM neural network.

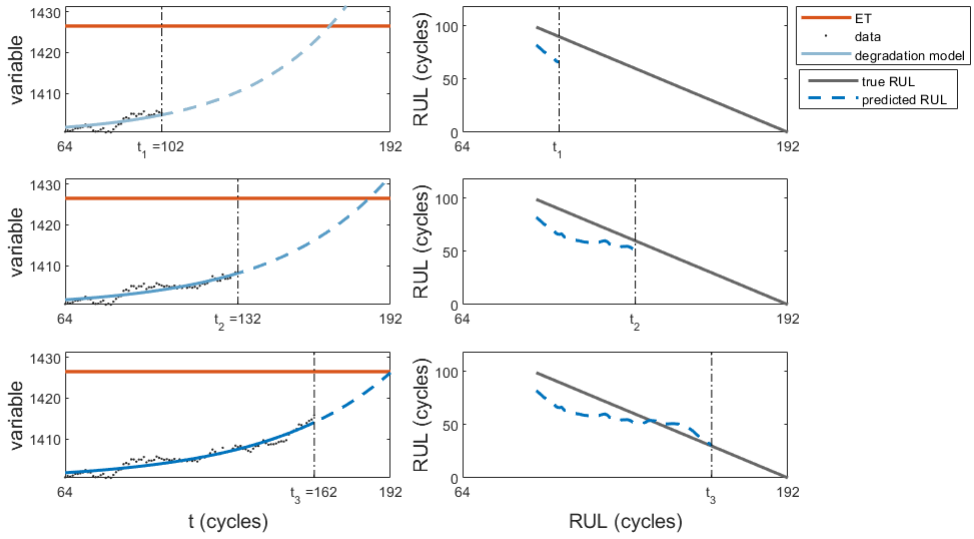


Fig. 5.19: On-line fitting and prediction for variable 4 of engine 1

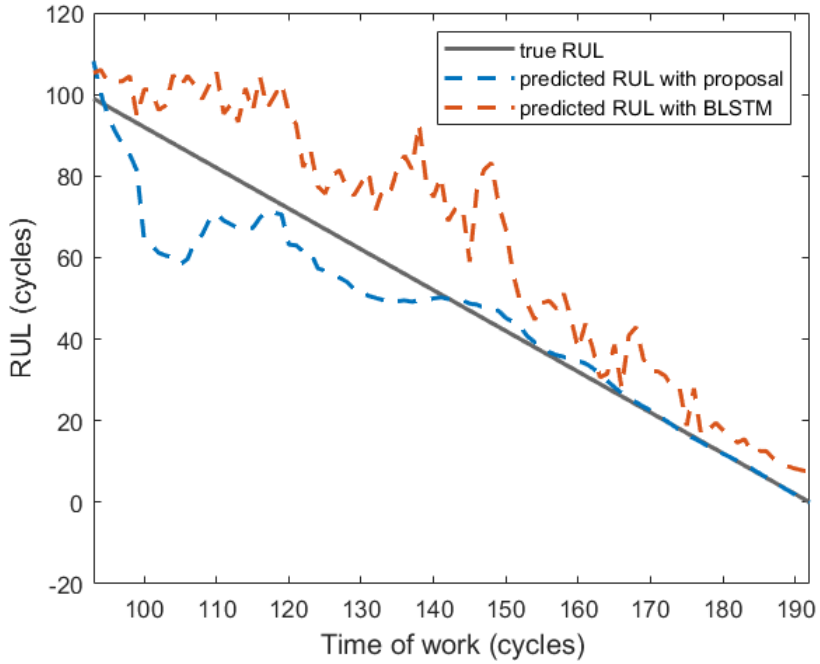


Fig. 5.20: RUL prediction for engine 1

5.5 Conclusion and future work

In this chapter, a new procedure is proposed to create RUL prediction models. To evaluate its effectiveness, the C-MAPSS data set is used, with exponential degradation models. The prediction model obtained is compared with a DNN-based model because of its widespread use in prediction tasks. Pre-

diction models are compared using a performance metric, RWMSE, which is proposed to evaluate RUL prediction models. This metric allows characterizing the evolution of the prediction error of the models and links the prediction of the RUL with the planning of maintenance tasks. The following conclusions can be stated based on the experimental results.

1. The prediction model adapts to the current conditions of the system during the degradation process, allowing to increase the prediction precision as the degradation progresses.
2. Constraints based on SVDD models allow accurate modeling of the historical behavior of the degradation process, which reduces the prediction error at the beginning of the degradation process when there are few measurements available.
3. Multivariable analysis determines the influence of each variable on the prediction of RUL and measures the importance of each variable for the RUL prediction process.

In future research, the procedure will be applied to other systems that reflects exponential behavior degradation. In addition, it will be extended to systems with other behaviors during their degradation process. Precision metrics for prognostics will be also analyzed.

Acknowledgements

Adriana Villalón-Falcón, Alberto Prieto-Moreno, and Orestes Llanes-Santiago acknowledge the financial support provided by Project No. 27, National Program of Research and Innovation ARIA from the Ministry of Science, Technology and Environment (CITMA), Cuba and Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE.

References

Ali, J. B., Chebel-Morello, B., Saidi, L., Malinowski, S., and Fnaiech, F. (2015). Accurate bearing remaining useful life prediction based on weibull distribution and artificial neural network. *Mechanical Systems and Signal Processing*, 56:150–172.

Ali, J. B., Saidi, L., Harrath, S., Bechhoefer, E., and Benbouzid, M. (2018). Online automatic diagnosis of wind turbine bearings progressive degradation under real experimental conditions based on unsupervised machine learning. *Applied Acoustics*, 132:167–181.

Branch, M. A., Coleman, T. F., and Li, Y. (1999). A subspace, interior and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23.

Bregon, A. and Daigle, M. J. (2019). Fundamentals of prognostics. In *Fault Diagnosis of Dynamic Systems*, pages 409–432. Springer.

Coble, J. and Hines, J. W. (2009). Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Annual conference of the prognostics and health management society*, volume 27.

Coble, J. B. (2010). *Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters*. PhD thesis, University of Tennessee.

Coleman, T. F. and Li, Y. (1996a). An interior point trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445.

Coleman, T. F. and Li, Y. (1996b). A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058.

Di Maio, Franceso an Turati, P. and Zio, E. (2016). Prediction capability assessment of data-driven prognostic methods for railway applications. In *Proceedings of the third European conference of the prognostic and health management society*.

Dong, Q., Cui, L., and Si, S. (2019). Reliability and availability analysis of stochastic degradation systems based on bivariate wiener processes. *Applied Mathematical Modelling*, 79:414–433.

Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., and Zhang, H. (2018). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183:240–251.

Feng, J., Kvam, P., and Tang, Y. (2016). Remaining useful lifetime prediction based on the damage-marker bivariate degradation model: A case study on lithium-ion batteries used in electric vehicles. *Engineering Failure Analysis*, 70:323–342.

Guo, L., Li, N., Jia, F., Lei, Y., and Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, 240:98–109.

ISO (2015). ISO 13381-1:2015 - Condition Monitoring and Diagnostics of Machines - Prognostics - Part 1: General Guidelines. Technical report, International Organization of Standardization.

Kim, N.-H., An, D., and Choi, J.-H. (2016). *Prognostics and health management of engineering systems: An introduction*. Springer.

Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., and Dybala, J. (2016). A model-based method for remaining useful life prediction of machinery. *IEEE Transactions on Reliability*, 65(3):1314–1326.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., and Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104:799–834.

Li, N., Lei, Y., Lin, J., and Ding, S. X. (2015). An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Industrial Electronics*, 62(12):7762–7773.

Li, X., Ding, Q., and Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering and System Safety*, 172:1–11.

Liao, L., Jin, W., and Pavel, R. (2016). Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment. *IEEE Transactions on Industrial Electronics*, 63(11):7076–7083.

Liu, K., Gebraeel, N. Z., and Shi, J. (2013). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 10(3):652–664.

Niu, G. (2017). *Data-driven technology for engineering systems health management*. Springer.

Saidi, L., Ali, J. B., Bechhoefer, E., and Benbouzid, M. (2017). Wind turbine high-speed shaft bearings health prognosis through a spectral kurtosis-derived indices and svr. *Applied Acoustics*, 120:1–8.

Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1–9. IEEE.

Song, Y., Shi, G., Chen, L., Huang, X., and Xia, T. (2018). Remaining useful life prediction of turbofan engine using hybrid model based on autoencoder and bidirectional long short-term memory. *Journal of Shanghai Jiatong University (Science)*, 23(1):85–94.

Tax, D. M. and Dui, R. P. (2004). Support vector data description. *Maching Learning*, 54(1):45–66.

Wu, Y., Yuan, M., Dong, S., Lin, L., and Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla lstm neural networks. *Neurocomputing*, 275:167–179.

Yan, M., Wang, X., Wang, B., Chang, M., and Muhammad, I. (2020). Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. *ISA transactions*, 98:471–482.

Yang, B., Liu, R., and Zio, E. (2019). Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Transactions on Industrial Electronics*, 66(12):9521–9530.

Zeng, Z., Di Maio, F., Zio, E., and Kang, R. (2017). A hierarchical decision-making framework for the assessment of the prediction capability of prognostic methods. *Proceedings of the Institution of Mechanical Engineers, Part 0: Journal of Risk and Reliability*, 231(1):36–52.

Zhang, H., Chen, M., Xi, X., and Zhou, D. (2017). Remaining useful life prediction for degradation processes with long-range dependence. *IEEE Transactions on Reliability*, 66(4):1368–1379.

Zhang, H., Mo, Z., Wang, J., and Miao, Q. (2019). Nonlinear-drifted fractional brownian motion with multiple hidden state variables for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Reliability*, 69(2):768–780.

Zheng, S., Ristovski, K., Farahat, A., and Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95. IEEE.

Zhu, J., Chen, N., and Peng, W. (2018). Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Transactions on Industrial Electronics*, 66(4):3208–3216.

This page intentionally left blank